1999

Compute as Fast as the Engineers Can Think! Utrafast Computing Team Final Report

Robert T. Biedron

P. Mehrotra

Michael L. Nelson
Old Dominion University

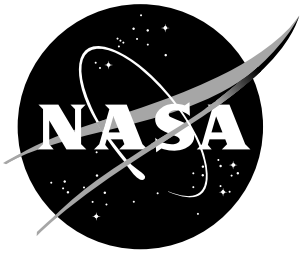
M. L. Preston

J. J. Rehder

See next page for additional authors

Authors
Robert T. Biedron, P. Mehrotra, Michael L. Nelson, M. L. Preston, J. J. Rehder, J. L. Rogersm, D. H. Rudy, J. Sobieski, and O. O. Storaasli

NASA/TM-1999-209715



Compute as Fast as the Engineers Can Think!

ULTRAFAST COMPUTING TEAM FINAL REPORT

R. T. Biedron, P. Mehrotra, M. L. Nelson, F. S. Preston, J. J. Rehder, J. L. Rogers,
D. H. Rudy, J. Sobieski, and O. O. Storaasli
Langley Research Center, Hampton, Virginia

September 1999

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

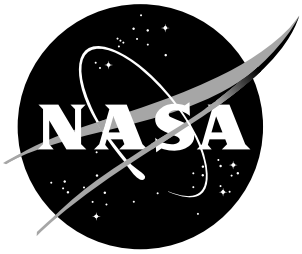
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-1999-209715



Compute as Fast as the Engineers Can Think!

ULTRAFAST COMPUTING TEAM FINAL REPORT

R. T. Biedron, P. Mehrotra, M. L. Nelson, F. S. Preston, J. J. Rehder, J. L. Rogers,
D. H. Rudy, J. Sobieski, and O. O. Storaasli
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

September 1999

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

This report is also available in electronic form at URL http://techreports.larc.nasa.gov/ltrs/

Preface

A committee whose roster appears on the title page wrote this report. The committee members were drawn from various NASA LaRC organizations. J. Sobieski served as the Committee Chairman. The committee's task was to examine the impact of the new computer architectures on computing in engineering in general, and more specifically on the computational support of the aerospace vehicle design process. It was formed in September 1998 by Douglas Dwoyer, Director of the LaRC Research and Technology Group, and completed its work in December 1998.

Abstract

This report documents findings and recommendations by the Ultrafast Computing Team (UCT). In the period 10-12/98, UCT reviewed design case scenarios for a supersonic transport and a reusable launch vehicle to derive computing requirements necessary for support of a design process with efficiency so radically improved that human thought rather than the computer paces the process. Assessment of the present computing capability against the above requirements indicated a need for further improvement in computing speed by several orders of magnitude to reduce time to solution from tens of hours to seconds in major applications. Evaluation of the trends in computer technology revealed a potential to attain the postulated improvement by further increases of single processor performance combined with massively parallel processing in a heterogeneous environment. However, utilization of massively parallel processing to its full capability will require redevelopment of the engineering analysis and optimization methods, including invention of new paradigms. To that end UCT recommends initiation of a new activity at LaRC called Computational Engineering for development of new methods and tools geared to the new computer architectures in disciplines, their coordination, and validation and benefit demonstration through applications

Vision

Computing that underlies the engineering design should be so capable that it no longer acts as a brake on the flow of creative human thought in the design process. The capability of the human mind to formulate concepts and digest data rather than the computing would then pace that process.

Compute as fast as the engineers can think!

Table of Contents

0. Glossary

CAS	Computational AeroSciences, a subset of HPCCP.
CE	Computational Engineering
CET	CE Team
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-Shelf
EDOF	Elastic Degrees of Freedom
ETTS	Elapsed Time to Solution is the wall clock time elapsed from input to output of a computing job.
FEA	Finite Element Analysis
FEM	Finite Element Model, the input to FEA.
FLOPS	Floating Point Operations per Second (see note below on computing speed)
HPCCP	High Performance Computing and Communication Program, a federally-funded research program.
HSCT	High Speed Civil Transport, a supersonic transport aircraft.
IIOP	Internet Inter-ORB Protocol
IT	Information Technology
ISE	Intelligent Synthesis Environment, a new NASA research and technology development program
NRA	NASA Research Announcement
PDE	partial differential equations
RLV	Reusable Launch Vehicle.
SAND	Simultaneous Analysis and Design
UCT	Ultrafast Computing Team

Note on computing speed, measured in FLOPS:

- CPU or processor speed quantifies the processing capacity of an isolated processor.
- peak speed of a multiprocessor computer equals the processor speed times the number of processors
- effective speed of a multiprocessor computer equals the peak speed reduced (usually very substantially) by the data transfer overhead.

Other terms used locally are defined where they first occur.

1.0 Introduction and Executive Summary

This document summarizes the deliberations, findings, and recommendations of the Ultrafast Computing Team (UCT). The UCT was formed in September 1998 at the instigation of Douglas Dwoyer, Director of the LaRC Research and Technology Group, and concluded its work in December 1998.

The UCT charge was to examine the impact of the new computer architectures on computing in engineering in general, and more specifically on the computational support of the aerospace vehicle design process. There were two reasons to be concerned about the impact. One reason was the mounting evidence from ongoing projects, at NASA and in the aerospace industry, that the computing capabilities delivered by the current combination of the computer hardware and the engineering analysis and optimization methods fall far short of what is needed for reliable and timely design processes accounting for both the vehicle physics and the vehicle life-cycle requirements. The other reason was a recognition that the computer software and hardware technology is rapidly changing toward a massively heterogeneous environment of diverse machines ranging from a single processor palmtop to a supercomputer containing thousands of processors, all unified in intelligent networks. That new trend offers a tremendous opportunity to lift the computational support of engineering design to a new, radically higher level of capability.

To fulfill its charge UCT began (Section 2) with two user scenarios, one for a supersonic transport and one for a reusable launch vehicle. Each scenario described a particular design situation that called for certain computations to be done in support of design decisions, and was used to derive computational requirements for major disciplines in terms of computing speed, time-to-solution, storage requirements, etc. When these requirements were compared to the capabilities currently available it was found that further progress by several orders of magnitude is needed to make the computing underlying the design process so capable that it would no longer act as a brake on the flow of creative human thought.

Given the broad spectrum of the vehicle technologies involved in the user scenarios, the vast dimensions of the computer technology, and the limited time and resources for the task, UCT had to be selective in its focus. Consequently, the report concentrates on the user requirements and computer technology aspects deemed most important for its purpose and deliberately omits a number of issues in the requirements and computer technology areas that were judged to be of lesser importance.

Accordingly, the UCT attention focused (Section 3) on the assessment of the trends in the computer hardware and software, distinguishing in the latter the infrastructure software from the engineering analysis and optimization methods. The assessment determined that further progress in the single processor speed will probably continue until it will eventually be slowed down and even arrested by the fundamental limitations of the silicon technology. Parallel to that, the trend toward

massively parallel processing will gather momentum to the extent that predictions of machines with a million processors begin to appear in literature, while machines with about 10,000 processors already operate at Sandia and Livermore Labs. The million processors machines will be placed at the high end of the spectrum of computers that spans a large variety of computer architectures, and technologies in both hardware and software , beginning with a palm top at the lower end.

From the assessments of the computational requirements and the computer technology trends, UCT inferred (Section 4) that the ultimate goal of making the computing underlying the design process so capable that it no longer acts as a brake on the flow of the creative human thought is possible by an exploitation of both the advances in a single processor performance and the multiprocessor parallelism. However, that exploitation will require matching the variety of computing jobs to the spectrum of computer architectures. That matching will require development of new, innovative engineering analysis and optimization methods for effective utilization of massively parallel processing. In some instances these new methods may have to be so new and different that they will constitute new computing paradigms.

UCT concluded (Section 5) that Government seed money and leadership is necessary for development of the above new methods because that segment of the market is too small and risky to attract commercial vendors. Consequently, UCT recommends:

> **the initiation (Section 5) of a new activity at LaRC, provisionally called Computational Engineering (CE). The CE should have two components: (1) the interdisciplinary activity to identify needs and opportunities resulting from the new computer architectures and the engineering design processes, to coordinate the disciplinary developments, and to carry out validation and benefit demonstrations by applications in aeronautics and space; and (2) a disciplinary component consisting of activities in all the disciplines to gear the disciplinary tools and methods to new computer architectures.**

The recommended implementation of CE may occur through the present CAS Team with a broadened charter and funding drawn from the continuing HPCC Program and the new ISE Program. In fact, the CE development has a potential of becoming the principal support LaRC will be providing to the ISE Program.

Considering the long lead time and magnitude of the resources needed for bringing the new methods for CE from inception to practical use, the time to start the recommended CE activity is now.

2. 0 User Requirements

When software developers plan for a future product, e.g., a word processor, they usually begin with writing a number of scenarios to scope out what users might want to do with the product. It is now well established that such User Scenarios are a more effective way for deriving the requirements and specifications for the product than trying to write a list of the product functions and requirements in abstract. We have adopted that approach to scope requirements for ultrafast computing and considered two scenarios that appear verbatim in the Appendix and are summarized in this section.

The first scenario is for a hypothetical supersonic transport design to be performed in 2011. It was developed jointly by the LaRC and LeRC HPCCP/CAS teams in the summer of 1997 for the purposes of the Information Power Grid planning (an ARC initiative). The second scenario was developed at LaRC in November 1998 by a Fast, Efficient Design Tools Team whose task was to plan for tools development for design of a Reusable Launch Vehicle in 2020.

2.1 User scenarios - a summary of the user scenarios in the appendix

This section presents summaries of the two user scenarios described in appendices A.1 and A.2.

Assumption: Tens of thousands processors are available and engineering analysis codes are capable of exploiting that parallelism.

Scenario 1: Multi-company design of a supersonic transport aircraft

A supersonic transport aircraft is at an early preliminary design stage. The multi-company partnership has one week to chose between two proposed alternative configurations. The choice will hinge on small differences of large numbers in multidisciplinary trades, e.g., drag vs. weight, hence a high fidelity of analyses is important. There will be a preponderance of previously generated data available through an efficient data retrieval system. There is a separate engineering group (geographically dispersed across the US) for each of the following: aerodynamics, structures, propulsion, life-cycle economics, and aircraft performance; and each group belongs to a different company. The performance objective to be maximized is the range for a given payload under all the disciplinary constraints.

Scenario 2: Multi-company design of a reusable launch vehicle

A reusable launch vehicle is at an early preliminary design stage. The multi-company, geographically dispersed partnership has one week to chose between two

proposed alternative configurations. The mission profitability, and the corresponding return on investment (ROI), will hinge on small differences of large numbers in multidisciplinary trades, e.g., structural weight vs. structure cost. This trade-off must account for the structural weight effect on the propulsion required and its influence on the vehicle gross weight and total cost, hence a high fidelity of analyses is important. The performance objective to be maximized is the return on investment over the life of the vehicle, including the assumptions of 10 years and 36 launches per year. The return on investment must account for the life-cycle economics.

The task is made more complicated by the discovery in the aeroelastic analysis that in one of the configurations, the structural displacements of the vehicle nose is excessive in a way it displaces the shock wave impinging on the engine inlet. An attempt to fix the problem by structural optimization led to excessive structural weight penalty, therefore, an alternative of relocating the engine to make it less sensitive to the nose-generated shock wave is under consideration. However, to assess fully the benefit of that alternative, one has to return to the conceptual stage in order to modify the configuration.

Time Constraints and Computing Requirements derived from the above scenarios

Time Constraints

- Complete the round of concurrent disciplinary operation in less than 60 minutes.
- Complete the collaborative system-level optimization and assessment of the results before committing to the next round of disciplinary optimizations in less than 90 minutes.
- Sustain a pace of 4 to 5 cycles in a working day, each cycle entailing first two items above.
- Obtain answer to a minor "what if" question in less than 10 seconds
- Obtain an answer to a major "what if" question in less than 2 minutes.

1. Flexibility to "shift gears" (computing agility) between codes of different fidelity to trade computing cost for accuracy, to return from the detailed stage to the conceptual one as the need requires, and to change from one MDO method to another

Computing Requirements

- CFD, Navier-Stokes-level, including rarefied atmosphere reactive flow, entire configuration analysis with derivatives of output to input turn-around, less than 1 minute
- Structural analysis, FEM-level, entire airframe of about 100K EDOF, 1000 loading cases, with derivatives of output to input turn-around, including heat transfer and hot-structures, less than 10 sec.

- Propulsion, including solid booster, electromagnetic propulsion, and air-breathing propulsion, 1 minute to match surrounding simulation optimization time.
- Simulation of dynamic phenomena, e.g., the vehicle handling, close to the real time scale.
- Modeling of the life-cycle elements other than the vehicle physics, taking into account the predominantly discrete and statistical nature of these elements in elapsed times comparable to those achieved for the vehicle physics modeling. This includes the analysis of cost and revenue necessary for assessment of the ROI.
- Other disciplines turn-around – nearly instantaneous.
- Modeling of the management of the design process in elapsed times comparable to those achieved for the vehicle physics modeling.
- Use of intelligent agents for continual monitoring of the process, information gathering, warnings about unexpected conditions arising in the process, and making routine decisions.
- Simulation of dynamic phenomena, including real-time refreshing of at least a part of the data.
- Visualization and Virtual Reality capability to support real time movies to display dynamic phenomena: at least 24 frames/sec on high-resolution screens.
- Immersive capability enabling a virtual walk throughout the mission architecture and its execution in time; peeling off the external parts of the vehicle to look inside, and invoking analysis by touching the part of the vehicle and pointing to the computing tool to be applied to that part.

2.2 Computing requirements for two key disciplines: Computational Fluid Dynamics and Computational Structural Mechanics

The complete scenarios for the supersonic transport and RLV involve all the aerospace disciplines. Estimating the requirement for all of them would exceed the report scope. Therefore, the estimates were limited to CFD and CSM singled out as the known greatest consumers of the computing resources in aerospace vehicle design. These estimates probably cover about 50 % of the total and are sufficient to establish the orders of magnitude involved.

The goal of rapidly computing a Navier-Stokes solution on a complete aircraft configuration, including sensitivity derivatives, will require substantial computing resources. What constitutes a "complete" configuration is subject to interpretation, but at a minimum should include wing, tails, fuselage, nacelles, and control surfaces. The *minimum* number of grid points required for a solution on such a configuration would be on the order of 10^7; on the order of 10^8 grid points would be required for a well-resolved flow field, and thus is the basis for the requirements presented below.

Structured-grid flow solvers (http://science.nas.nasa.gov/~faulkner/cfd_perf.html) require significantly less memory and floating point operations than unstructured-grid solvers. However, the ability to easily generate unstructured grids around complex configurations means that unstructured solvers [Mavriplis 1998] are the tools of choice for the rapid design envisioned in the scenario.

Correspondingly, the structural analysis of an airframe, for a supersonic transport or a reusable launch vehicle, is expected to require a finite element model of the order of 100,000 elements, 300,000 nodes, 900,000 EDOF, (average 3 EDOF per node), and 1,000 loading cases. For dynamic analysis purposes the number of vibration modes needed for flutter analysis is of the order of 50 repeated to account for the symmetric and antisymmetric boundary conditions.

2.2.1 CFD requirements

CPU requirements

A widely-used structured-grid Navier-Stokes solver was recently documented as requiring approximately 8600 floating point operations/grid point/multigrid cycle. Recently published data for an unstructured solver using the same basic algorithm indicates approximately 49,000 floating point operations/grid point/multigrid cycle are required for the equivalent computation on an unstructured grid. Assuming that 500 multigrid cycles are required for a converged solution, then roughly 10^{15} operations are required for the function evaluation on an unstructured grid of 10^8 points.

For derivative evaluation, assume that an adjoint formulation is used, so that any number of derivatives can be obtained with a solution to the adjoint equations. Although in principle the adjoint equations can be solved in approximately the same operation count as the flow equations, experience suggests that depending on the formulation used (continuous or discrete) up to 5 times more operations may be required as compared to the function evaluation. If automatic adjoint tools are used, the factor can be as much as 20. Assuming a factor of 10 as an average, evaluation of derivatives for the complete aircraft will require on the order of 10^{16} operations for an unstructured grid with 10^8 points.

Therefore, to obtain the required function and gradient evaluations in approximately 1 minute will require a machine capable of a sustained rate (effective speed) of approximately 10^{15} floating point operations per second. It is worth noting that in order to meet the computing requirements in the scenario, such computing speeds must be routine, not one-of-a kind computations.

The scenario for RLV requires analysis of a reactive flow in rarified atmosphere that is likely to amplify the above estimates by a factor of 4, assuming an 11-species air

model, and non-equilibrium gas computation. Moreover, unsteady aerodynamic analysis required for flutter and control is likely to add another multiplier of the order of at least 10, more if animation of time-dependent phenomena is needed. Thus, the estimates developed above should be regarded as lower bound.

Memory and storage requirements

Current unstructured solvers require between 250 and 500 words per grid point (compared to 25-50 for structured solvers). The higher bound implies $5x10^{10}$ words for a grid of 10^8 points, or roughly 400 GBytes of memory for double precision arithmetic. The storage requirement for an unstructured grid of this size would be roughly 50 GBytes; storage of the solution would require 6 or 7 times this amount, depending on the turbulence model. These estimates increase by about a factor of 10 for reactive flow in rarified atmosphere.

2.2.2 CSM requirements

The Finite Element Analysis (FEA) of a 900,000 EDOF structure leads to linear equations whose matrix of coefficients is 900,000 x 900,000 but is very sparse. The scarcity order is of 1%. The individual loading cases appear as separate right hand side vectors, an ideal opportunity for coarse-grained parallel processing.

The volume of output can be estimated by assuming
- up to 6 displacement values per node
- 3 stress values at four corners and the center of a membrane element.
- 10 strength and buckling constraints per element
- 20 composite plies in a membrane element.

Under the above assumptions the number of displacements to be output for 1,000 loading cases is 900,000 x 1,000 = $.9 * 10^9$. The number of stress values is 100,000 x 3 x 5 x 20 x 1,000 = $30 * 10^9$. The number of strength constraints is NSC = 100,000 x 10 x 1,000 = $1 * 10^9$.

The number of arithmetical operations for factoring (NAOF) of a banded matrix of the load-deflection equations is (EDOF)/2 * $(\text{Bandwidth})^2$. The estimate of the Bandwidth (only the symmetric half) is 4,500, hence NAOF = $9 * 10^{12}$. The preprocessing of the load-deflection equations at least quadruples the above. Forward and back substitution requires NAO of about 1% of NAOF per loading case and another NAO for stress postprocessing about equal to that for preprocessing but multiplied by the number of loading cases. Hence, the estimate for the total number of arithmetical operations for the subject structure is TNAO = NAOF*((4 + ((1/100)+4) * 1,000) = $36 * 10^{15}$.

Sensitivity analysis for NDV design variables requires repetition of the above the number of times which varies from (NDV+1) for finite differencing or automatic

differentiation, to about 1/3 (NDV) for the direct analytical method. If the adjoint sensitivity method is used the forward and back substitutions have to be repeated the number of times equal to the order of the number of critical loading cases, NCLC, and the stress postprocessing needs to be repeated the number of times equal to the number of the critical constrains, NCC.

Consequently the TNAOAS, the total NAO for analysis with sensitivity, may be given for finite differencing or automatic differentiation

TNAOAS1 = TNAO * (NDV + 1)

for the direct analytical method

TNAOAS2 = TNAO + TNAO * NDV/3

and for the adjoint sensitivity analysis method

TNAOAS3 = TNAO + NAOF * (1/100)*NCLC + NAOF * 4 * NCC

Assuming NDV 5000 , NCLC = 10, and NCC =100, the estimates are TNAOAS1 = 180 * 10^{18}; TNAOAS2 = 60 * 10^{18}, and TNAOAS3 = 4 * 10^{16}. To execute the above in 1 minute would require a machine capable of a sustained performance of the order of 10^{15} to 10^{18} FLOPS.

It is notable how strongly the above estimates depend on the choice of method and that the lowest estimate of 10^{15} FLOPS agrees with that derived for CFD in the preceding section.

Memory and storage requirements

The storage requirement is dominated by the stiffness matrix and the matrix of the constraint gradients, the Jacobian, needed in gradient-guided optimization. The stiffness matrix, symmetric with 1% scarcity, requires the number of words in storage equal to 0.01 * ($EDOF^2$-EDOF)/2 + EDOF = 4 * 10^9.

Current equation solvers require between 350 and 700 bytes of memory for each degree of freedom (equation). The higher the number and complexity of structural elements connecting the degrees of freedom, the more memory is requiredfor solution per equation. The 53 second solution of a 551,585 equation aircraft model requires 2 Gigabytes of memory (14 GB of real memory is available on our Origin 2000). Unlike CFD codes, memory to solve large-scale structural analyses may be either real or “virtual” using current “out-of-core” (using disk for memory) solution techniques.

Although most structures codes have solvers with “out-of-core” capability, such operations involve a serious performance “hit” often slowing down solutions by an

order of magnitude compared to use of "real" memory. Thus, memory is not, per se, a limiting factor to solve large structural problems. However, more memory speeds up the solution. With memory capacities increasing according to Moore's law, structures solutions will benefit by less reliance on "out-of-core" solutions and significantly faster solution times. With typical departmental computers with 14GB of real memory now are capable of solving 4 million equations in memory ans virtually unlimited sizes using virtual memory. In 10 years, following Moore's law projections for memory growth, typical departmental computers will have 300MB memories and be capable of solving 85 million equations directly in memory and virtually unlimited sizes "out-of-core".

Optimization memory and storage requirements are driven by the dimensions of the Jacobian matrix that contains the derivatives of constraints with respect to design variables. To estimate the dimensions of that matrix assume the number of design variables needed to describe the aerodynamic shape, and the cross-sectional dimensions to be of the order of 100 and 5,000, respectively (from the sources reviewed in [Sobieszczanski-Sobieski and Haftka 1997]). The Jacobian dimensions are (number of design variables) * (number of constraints). For the data introduced in the foregoing, this yields 10^9 x 5,000. This is, generally, a full matrix so its content that needs storage is of the order of $5 * 10^{12}$.

2.2.3 Coupling of CFD and CSM

CFD and CSM exchange the data on aerodynamic loads and structural deformation. Assuming 10 flight conditions, the full extent of the data to be so exchanged is of the order of 10 x (number of FEM nodes = 300,000) = $3 * 10^6$. Condensation techniques may be used to reduce the volume of data by 100 to 1,000, but they are problem dependent and introduce uncertainty. The data exchange occurs at least once for the combined CFD and CSM analysis. The analysis is iterated, typically, 5 to 7 times for nonlinear CFD.

2.3 Optimization and other requirements

Optimization requires repetition of analysis. The use of modern approximation methods and decomposition procedures has decoupled the number of the repetitions from the number of variables. The current practice, surveyed in [Sobieszczanski-Sobieski and Haftka 1997], indicates that the number is of the order of 50, however each analysis must include sensitivity analysis (or an equivalent effort for generation of a response surface or training a neural net). The CPU time added for the sensitivity analysis can be estimated for NDV design variables by

multiplying the analysis time by the factor of N, where N varies from 0.3 NDV, for an analytical method, to 2 NDV, for finite differencing or automatic differentiation.

Thus, for NDV = 5,000, the analysis cost embedded in the optimization of a combined CFD & CSM problem is of the order of the single combined analysis cost multiplied by 50 x 5,000 x (0.3 to 2) = (7.5 to 50) * 10^4. However, a dominant portion of the gradient computation for each design variable is independent of the other design variables, hence an opportunity to employ a very large number of concurrently operating processors.

The repetitive analysis requirement is not limited to optimization. It is also the key requirement in visualization of dynamic phenomena to support 24 frames/second animation, and in the step-by-step simulation of such phenomena in real time. Computations that support these types of applications usually mix reanalysis with retrieval of previously generated data and employ sophisticated data compression schemes, therefore, reliable estimates for additional computing requirement are problem-dependent and difficult to formulate. However, it is reasonable to expect that they are substantial, both in terms of computing speed and time for data retrieval. This opinion is substantiated in Section 4.5 on Human Interface and in Appendix Section A3 on Real Time Simulation and Visualization.

2.4 How fast is fast enough?

Acknowledging that one of the principal drivers of the computer technology progress is the demand for ever faster computing, it is useful to ponder the question "how fast is fast enough?". To examine that question, consider the following major operations that can be discerned in either of the above scenarios:

1) Human decisions regarding the relatively few but very important aspects of design where choices cannot be made by automated computing.

2) Analysis combined with optimization regarding the large number of detail-level design decisions that can be automated within the limits set by engineers. Warnings must sound whenever the solutions press against these limits. Short elapsed time is the critical requirement in this operation.

3) The above operations are interleaved.

4) Occasionally, surprise new information is discovered at a late design stage that compels one to return to earlier stages. In the extreme, one may have to return to the conceptual stage or even to the requirement formulation. The return should

be encouraged by making it easy, because the design leverage at early stages is usually much greater than at the later stages.

Item 4 above suggests computing agility, a term for an ability to move expediently between the level of details in the design process, as a requirement almost as important as the computing speed.

The computing speed itself is interpreted here as a variable that shortens the elapsed time-to-solution (ETTS). It is important to note that the computing speed is merely a measure of capability that the user may utilize either as a means to reduce ETTS, or to trade ETTS for accuracy, or to explore more design options in the calendar time allotted for the particular design task.

If the user opts entirely for a shorter ETTS, the progress is measured in discrete chunks. If a particular task typically requires an overnight 16 hour run, a reduction to 8 hours will hardly be perceived as a meaningful improvement because the results will still be coming back the next day. In contrast, a reduction to 2 hours would be a meaningful improvement as it would enable one to evaluate the results and to make decisions about three times a day.

By the same token, the ultimate goal should be to reduce ETTS for even a major problem to the order of a second, the time interval that would support the continuity of the train of thought in a creative pursuit of a better design. When that goal is achieved, the human mind will pace the design process, not the computer. Note that this is precisely the capability of the present day word processing. The length of time needed to write this report depends on this writer 's ability to compose sentences, the S/W & H/W of the word processor has a comfortable speed margin beyond that ability. That should be the goal to be ultimately reached in support of engineering design. This ambitious goal can only be attained by the synergy of new methods and new computer architectures. To put it succinctly:

"Compute as fast as engineers can think!"

To translate the above goal into a speed-up of the current computing capability, consider the requirement of 10^{15} FLOPS for the effective processing speed formulated for the CFD and CSM analysis problems, including sensitivity, in section 2.3.1 under an assumption of ETTS = 1 minute. Considering 50 analyses (with sensitivity) to converge the aircraft configuration optimization, and asking for ETTS = 1 second for that task would **require 50 * 60 = 3,000 greater speed, that is 3 * 10^{18}.** It is important to emphasize at this point that the above FLOPS estimates refer to the effective computing speed rather than to the peak speed.

3.0 Trends in Computing

With the requirements outlined in the previous section in mind, the UCT examined the projected technology trends, which are summarized in this section.

3.1 Current state of the field and predictions based on Moore's law

The most widely used measure of computer performance projections are based upon Moore's Law (founder of Intel) which projects a factor of four increase in computer chip performance every three years. Similarly, an increase in memory density has been experienced. The Semiconductor Industry Roadmap [Anon. 1997] is based upon this continuing for about the next ten years. Beyond about 2007, the present technology will not support this growth (the X-ray lithography on which the chip manufacturing technology is based will reach its fundamental resolution limit), but individuals in the chip business have confidence that progress will be maintained, based upon "… it always has, it will continue." One may note that the NASA ARC initiative called Information Power Grid (IPG) is based, in part, on the expectation that a "brick wall" preventing further increases of single-processor speed, [Feiereisen 1998], will be encountered by about 2010.

Based upon this rate of doubling every eighteen months, the growth assured by 2007 will be a factor of 64 in performance and memory density. The speculative growth by about 2013 could be a factor of 256 beyond 1998 performance. A major break-through could change this picture. Speculations abound on such breakthroughs, e.g., quantum computing, chemical computing, etc., however, the time to the application readiness for any of these inventions cannot be predicted with any degree of certainty. Therefore, these speculations are left out of the scope of this report.

The two scenarios (Appendices A.1 and A.2) interpreted in Section 2.0 represent real requirements that are not being met now. They substantiated the estimate of the effective computing speed of the order of 10^{18} FLOPS as given at the conclusion of Section 2.4.

The currently available processor speed approaches the order of 10^9, hence the above estimate suggests that further speed-up by 10^9 is needed. Assuming that the Moore's law, doubling the processor speed every 18 month, continues unabated, it would take 45 years to achieve the required speedup. If a machine is built using 10^6 of the fastest, currently available processors and operated using a solution algorithm with zero overhead for interprocessor communication, it would deliver the effective speed of 10^{15} FLOPS, the peta-FLOPS level but still three orders of magnitude short of the projected requirement of 10^{18}. To make up for that shortfall, the individual processor speed needs to increase by the factor of 1,000, the speed-up likely to be provided by the Moore's law in about 15 years.

Of course, the data and calculations on which the above estimates rest are subject to a wide margin of uncertainty. Acknowledging that, the above order of magnitude evaluation indicates that **the computing with the effective speed that will make even a very large problem solution perceived as instantaneous is attainable in one to two decades partially by the progress in a single processor technology and partially by massively parallel processing. The caveat is that this will require development of methods that can effectively utilize a very large number of concurrently operating processors with almost no overhead for interprocessor communication. A new generation of methods will be required to do that so that the machines with very large number of processors are utilized at their full potential**.

3.2 Trends in computer architecture

In the not too distant past there was only one basic computer architecture, the classical model of a single CPU. In contrast to that today and for the foreseeable future the users will have a choice of an entire spectrum of various computer architectures, ranging from a single processor palmtop to supercomputers that employ massively parallel architectures. These architectures vary in many respects, e.g., different memory hierarchical arrangements, and many innovative ways of connecting processors with their dedicated memory blocks. The spectrum will be augmented with intelligent networks capable of making an assemblage of dissimilar machines appear as a single virtual computer to the user.

For several years now, the computer and component business has been operating in a commodity market mode. The mass market is driven by requirements that have broadened in scope and departed from the prior needs of users. Commercial and home use have diversified tasks, changing the requirements and driving components in new directions. Examples of this are more use of audio and video, more communications and storage. The mass market drives the availability and specifications for components. Economics trumps technology.

In earlier years, scientific computing was the driving force and had sufficient market clout to get its needs to control the availability of suitable components. Now, the scientific market is a very small percentage of the total. Thus, technical computers will have to be designed using commodity parts. Up to recently, this has produced suitable components. It is not certain if the mass market components will meet the demands of the scientific community in the next ten to twenty years. Thus, compromises may be forced on supercomputers.

The effect of these economic realities is evidenced by the fact that most hardware vendors have abandoned large distributed memory architectures and are manufacturing small shared memory machines (SMPs) with only 10s of processors since these can easily be sold in the mass market. Larger parallel machines for the scientific market are then constructed as clusters of SMPs using fast networks/switches as the interconnection medium. This trend is corroborated by

the sequence of teraflop machines (Figure 1) being installed for the Department of Energy's ASCI program (http://www.llnl.gov/asci/). Each of these machines consisting of thousands of processors, has been constructed by combining small SMP units to provide the required computing power. The issue with this approach is that such machines are significantly more complex and are thus more difficult to use effectively.

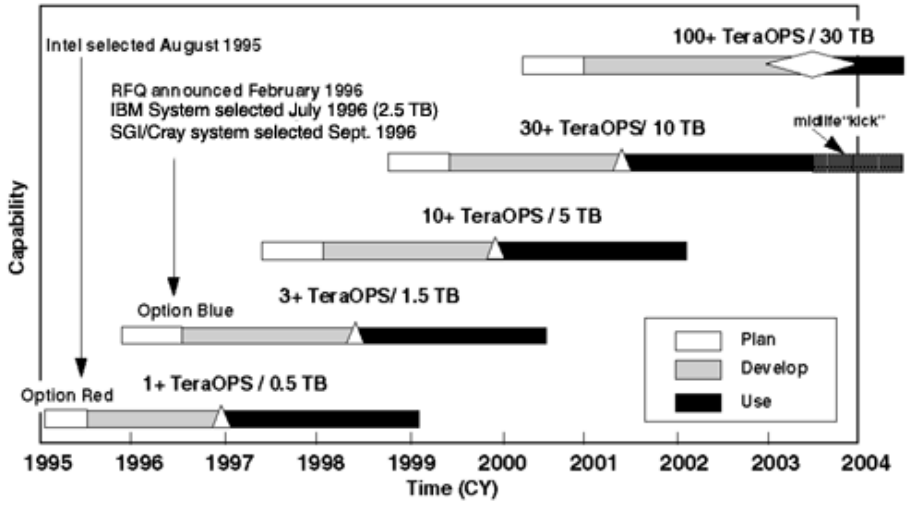


Figure 1 ASCI Multi-Teraflops Computing Roadmap

The impact of using COTS hardware also extends to software. The scientific community will need to increase its spending on development of applications and tools. As an example, in a multi-disciplinary study, one application will run separately much faster than another. All portions of the simulation should run approximately in synchronism. This will entail adjusting the number of processors between disciplines proportionately to the processing required. This means either doing this manually by experience or incorporating some adaptive system that will make the assignment, a complex task dependent on the logic decisions within the programs. Decisions will also be needed on the interactions at the interfaces of the different applications. Programming will become more complex in the multi-disciplinary environment and validation of software will then become extremely critical.

3.3 Trends in software

In the past few years, software technology has not kept pace with the advances in hardware architectures and systems. More success has been seen at the uniprocessor level than in the multi-processor arena. The technology developed for vectorizing compilers has proved to be fundamental for compilers targeting RISC architectures. However, cache management, which is critical for performance, is still a tricky issue.

On the other hand, research in automatic parallelization for parallel architectures has found only limited success in that, current compilers can produce good code for only moderate sized shared memory systems. For larger systems, and in particular for systems which exhibit memory locality, users have to resort to explicit parallelization to extract the full performance of the underlying architecture. The emergence of MPI (Message Passing Interface) for distributed memory systems has proved a boon for users in providing a stable and portable interface for a variety of multiprocessor architectures. A similar effort called OpenMP, is currently underway for shared memory systems. Other approaches, such as High Performance Fortran (HPF) which allow users to express the parallelism at a higher level, have only found limited success in niche areas. Driven by the realities of the commercial world, future multiprocessor architectures will tend to be small shared memory units clustered together to form larger parallel systems. Providing a programming interface for such systems is a difficult challenge, since it is not clear whether even users writing explicitly parallel code can effectively exploit the multiple levels of parallelism exhibited by these architectures.

Over the past few years, a large number of tools have been developed for debugging both logical and performance bugs in parallel programs. Users have adopted only a few of these tools, finding most of them to be too clumsy to use and not providing the needed information. In particular, it is not clear how such tools will scale with the increase of the number of processors envisioned in future multiprocessor systems along with the increase in the overall complexity in the architecture of such machines.

3.4 Trends in storage

Storage capacities are greatly increasing, but at a slower rate than computational power, primarily due to economic, non-technical constraints. The high performance computing community has now identified the widening gap in the ability to computer and create information, and the ability to store and access the information [Moore et al. 1998]. The magnetic disk as a mass storage device depends on packing density of the magnetic bits (dipole zones) for both the speed of access and volume of data. The current cutting edge of technology approaches 12 gigabits/sq. inch [Tristam 1998]. It is projected that when the density rises into the 20 to 40 gigabit range the magnetic zones will become too small to maintain their polarity that distinguishes between 0 and 1 and that polarity will become a random function of temperature corrupting the data.

To address physical storage needs for data intensive computing, there are 2 main classes of relevant projects: fully integrated and separately designed systems such as the High Performance Storage System (HPSS), and the commodity based extensions of existing storage systems. However, neither of these approaches attempt to address the requirement for higher level data management capabilities as called for in

[Moore et al. 1998]. HPSS [Coyne et al. 1993] is a joint project between DOE, NASA, IBM and other parties with an interest in high performance computing. HPSS is a hierarchical, distributed system that allows arbitrary storage components to be "plugged-in" to HPSS and managed through a separate suite of software. HPSS supports terabyte file sizes and systems with petabyte capacities. HPSS is also network centric (no hosting computer bottleneck), performs parallel file transport, etc. HPSS is one level above traditional file systems, but below traditional database technologies in terms of responsibilities. HPSS is independent of advances in storage mediums. More information on the HPSS project is available at the following Web site: http://www.sdsc.edu/hpss/.

In the commodity market, over the last 5 years prices for disk drives have fallen a factor of 2 per year, with area densities increasing 60% per year, and data rates increasing at 40% per year per year [Gibson et al. 1996, Grochowski and Hoyt 1996]. Although these trends exist in storage components, there is less industry confidence in the ability to sustain their advancement pace relative to microprocessors and Moore's Law. However, we believe that it is likely that these rates will continue, even if the underlying implementations shift. The first attempt to use commodity components was the Redundant Array of Inexpensive Disks (RAID) [Patterson et al. 1987] as a method of clustering commodity disk drives together for reliable, larger scale storage. Future trends in commodity usage will include moving away from RAIDs as they currently exist (the host computer is the throughput bottleneck), toward a fully distributed system similar to what is called "tertiary disks" [Talagala et al. 1998]: a 3TB system where many 8GB disks (370) are managed by 20 PCs, allowing greater capability for parallel access and reliability (See "Serverless Filesystems" [Anderson et al. 1998]). Cheaper disks made RAID possible, cheaper computers will make tertiary disks possible.

A reasonable and representative forecast for the Hardware Technology Trends related to future computers is contained in http://cs.berkeley.edu/~patterson/talks entitled: "Hardware Technology Trends and Database Opportunities" In summary, it projects:

- Disk Storage: Continued advance in capacity (60%/year) and bandwidth (40%/year)

- Networks: Limited by software protocols (Internal I/O bus is limiting factor)

- Memory: Capacity follows Moore's law 4x/3 years (60%/yr)

- Processor: Outpaces DRAM Performance by 50%/yr: 10('88), 100('94), 1000('99) SPEC performance doubling/18 months, embedded processor promise (Moore's law)

- Systems: "Greg's Law" database demand of 2x/9-12 months outstrips Moore's law and drives processor design forcing designers to stretch beyond Moore's Law for systems.

In conclusion, any advances we have seen in hardware, networking and systems in the past 10 years is likely to be surpassed in the next ten years.

3.5 Trends in networking

Networking technology progress arguably fares worse than processors or storage. Trends in networking can be analyzed in partially overlapping terms of Local Area Networks (LANs), Wide Area Networks (WANs), and the software and protocols that allow applications to send and receive data from the networks.

In terms of LAN capability, additional technology is currently available, but is not as widely adopted as would be expected [Stevens et al. 1997]. Most sites are in the 10Megabits/sec standard Ethernet, or the 100 Megabits/sec Switched Ethernet or FDDI range. Although 0.8 Gigabits/sec High Performance Parallel Interface (HIPPI-8) is available now, it is not widely deployed. 6.4 Gigabits/sec HIPPI-64 should be available in the near term.

For WAN capability, there are a number of initiatives now underway including Internet 2 (http://www.internet2.edu/) as well as the Next Generation Internet (http://www.ngi.gov/). Current testbeds are just now beginning to support 1 GB/s connections for WANs; it is unknown how multi GB/s WANs will be constructed to support future requirements Commercial attention to rapid broadband networks is dramatic as evidenced by recent ATT/TCI/@home and MCI/Worldcom mergers (takeovers). In his recent annual report to shareholders, the chairman of ATT stated ATT's plan to provide 3Mb/s service to 80% of households in the next 3 years. He also stated the future of ATT hinges on a rapid transition from long lines to broadband incorporating phone calls in the future within the internet broadband. Network competition is fierce, with broadband coming fast and cheap to the average consumer and a large proportion of ATT and other phone calls going over broadband (internet) rather than long lines in the next few years. This is echoed by Bill Gates in his recent addresses at Indiana University and the Manhattan Institute. Bill Gates is personally heavily invested in a worldwide high-speed network firm with numerous earth orbiting satellites (not Geosynchronous). It is not unreasonable to project that network speeds will be one of the fastest growing areas, surpassing even the Moore's law rate for CPU and memory over the next 10 years. Perhaps the largest barrier to utilizing both LAN and WAN technology is the current application-network interface. The operating system kernel and the networking stack are commonly identified as the largest single slowdown in networking. The popular networking protocols were designed in a low bandwidth, low latency context and such implementations are non-optimal in the current networking environment. For example, the Virtual Interface Architecture (VIA)

[von Eicken and Vogels 1998] is an emerging standard that features low latency and high bandwidth for small messages by allowing applications to send and receive network messages by bypassing the operating system and writing to a user-level network interface.

3.6 Trends in distributed computing

Recent advances in the Internet and in particular Web-based technologies have had a dramatic impact in the information technology arena. Initiated by work done by academics, most of these advances have been driven by the commercial world serving the needs of businesses. In particular, commercial forces are joining together to propose standard protocols and interface such as IIOP and CORBA, to allow disparate hardware and software resources to inter-operate with each other within this new paradigm. The introduction of Java has provided an important boost in this search for portability of programs. Another significant area of development is collaborative technologies which allow multiple users to simultaneously access and manipulate shared dynamic information.

These advances have not only changed the way that ordinary citizens access information and services but have also influenced the computing environment of scientists and engineers. Faster networks and widespread connectivity have allowed scientists to envision environments in which various data and computing resources can be brought together to synergistically solve a single large problem. This vision is central to several projects currently underway, including NASA's ISE program (http://cst-thor.larc.nasa.gov:80/ise/navigate.htm) and the proposed IPG (http://science.nas.nasa.gov/Groups/Tools/IPG/) activity, and the NSF supported efforts at the San Diego Supercomputing Center (http://www.sdsc.edu/) and NCSA (http://www.ncsa.uiuc.edu/) at the University of Illinois. Jini, a Sun R&D project, expands the power of Java by enabling spontaneous networking of a wide variety of hardware and software resources. These projects are focusing on the services and technologies required to integrate geographically distributed resources into a seamless computing fabric for solving large multidisciplinary problems.

4.0 Impact of Trends on Computing in Engineering Design and Research Directions

This section examines the impact of the computer technology trends on various aspects of computing in engineering design and stimulation of that impact on research directions. The aspects are: the classes of the computing problems; types of parallelism, algorithm research, distributed computing, and human interface with emphasis on visualization.

4.1 Classes of computing problems

The user scenarios call for solution of a multitude of engineering computing problems. It is axiomatic that these problems vary broadly in terms of the metrics such as the number of variables, number of arithmetic operations to achieve a solution, the volumes of input, intermediate, and output data, couplings among the variables, degree of interactive operation, degree of repetitiveness, human interface requirements, and the importance of the time-to-solution in the user perception.

The computer technology trend toward a broad spectrum of architectures poses a new problem that was nonexistent when the single processor machine was the only choice. The user's convenience, perceived efficiency, and real cost of solving a particular problem will depend, sometimes strongly, on the match between the problem metrics and the architecture of the machine.

Consequently, the following classification should underlie an effective planning for development of new methods in engineering computing:

- Clearly, a class of the computing problems may be identified that are solvable below the "real time" threshold on single processors of the existing or projected technology.

- Another class so solvable using coarse-grained parallelism of the "single code-different inputs" type.

- Another class so solvable using domain partitioning parallelism.

- Finally, it is likely that for a certain class of problems none of the above is adequate and new paradigms in conjunction with Massively Parallel Processing are needed.

The number of problems in each class and their relative importance is not known. Collection of at least approximate data for the above will be important for determining how the NASA-funded development and priorities should be distributed over the above classes of problems.

4.2 Types of parallelism

To exploit a machine with a large number of processors in engineering computing, one may choose among the following options:

1) Coarse-grained parallelization a code replicated and executed with different inputs
2) Coarse-grained parallelization by partitioning of the analysis domain
3) Fine-grained parallelization of an existing code
4) Fine-grained parallelization by recoding the solution algorithm from ground up
5) New paradigms to replace existing solution algorithms with new algorithms that are intrinsically parallel

Option #1 is the most straightforward to implement and is available immediately. It applies whenever a code is to be executed over and over again with different inputs. By replicating the code over as many processors as there are inputs, the elapsed time may be compressed to nearly the time of a single execution.

Opportunities for using that option are numerous. For example, a complete evaluation of the aerodynamic characteristics of an aircraft requires executing the same CFD code at various combinations of the Mach number and angles of attack. In airframe structural analysis, a large number of loading cases enters the FEA. The analysis involves a core part to be executed once for all loading cases, and postprocessing independent for each loading case. Thus the relative efficiency derived from a coarse-grained implementation of the postprocessing increases with the number of loading cases. A similar opportunity arises in the sensitivity analysis.

The advantage of Option #1, the coarse-grained parallelism, lies in the near absence of new coding and linear scalability in terms of the number of processors. However, the number of processors that can be concurrently engaged depends on the problem, and it is likely that in application to analysis it will not rise to O(4). The outlook is more optimistic for optimization where the methods of the genetic algorithm, response surfaces, and neural nets all call for very large numbers of independently analyzed points in the design space. Along the same line, the human interface to computing will benefit from animation or virtual reality supported by simultaneous generation of frames depicting a time-dependent phenomenon for a series of time steps.

Thus, one may expect that the option of coarse-grained parallelism will be used to advantage first and will buy time needed to explore the other options.

Option #2, coarse-grained parallelization by partitioning of the analysis domain, applies wherever a physical object is analyzed by discretization, e.g., the flow in 3D space around an aircraft analyzed by CFD on a 3D grid, or an airframe modeled as an assemblage of finite elements. In either example, the aerodynamic grid or the finite element mesh are partitioned into subdomains, each assigned to a separate

processor. That partitioning creates an opportunity to engage many processors concurrently and compresses the elapsed time. Unfortunately, the partitioning generates the need to reconcile the unknowns at the subdomain interfaces by forming and solving new equations that would not have existed had the problem remained in the original, nonpartitioned form. For fundamental geometrical reasons, the number of these interfaces quickly rises with the number of subdomains. This rise depresses the effective computing speed gained by partitioning, and eventually, the computing on the interfaces outweighs the gain from the concurrent subdomain processing. To see that drawback, think of a finite element model partitioned into substructures. When the number of substructures increases to the extreme that each finite element is treated as a substructure, the problem has circled back to the original finite element formulation and nothing has been gained.

It appears that Option #2 has a potential to engage many processors simultaneously—the number probably in the same range as in Option #1. It is important to observe that Options #1 and #2 are synergistic, because fine-grained parallelism may be used to speed up evaluation of each data point in a coarse-grained scheme. When combined, these two options may have a multiplicative effect on the number of processors that can be utilized concurrently. However, it does not appear likely that even with that synergy the number of processors in the range 10^5 to 10^6 could be effectively operated in parallel.

Options #3 and #4 require examination of the code at hand to identify the code segments that are independent of each other and, therefore, concurrently executable. Depending on the result of the examination, the choice may be to modify the code or to recreate it. Obviously, Options #3 and #4 have limited utility because they are labor intensive, error prone, and unlikely to achieve a uniform processor work distribution beyond relatively small number of processors. However, they may be used on occasion for speeding up Options #1 and #2.

Finally, it appears that Options #1 and #4, especially when combined, may be interim enablers of concurrent processing with the number of processors in the 10^3 to 10^4 range. To get beyond that range it will be necessary to pursue Option #5 that calls for inventing new paradigms.

4.3 Current state of the art in algorithmic and application research

Three factors determine how to obtain optimal solutions to a given application in the minimum time:

- the problem discipline specifics (structures, electromagnetics, acoustics and fluid mechanics)

- the matrix characteristics (dense/sparse, complex/real, indefinite/positive definite, non-symmetric/symmetric)
- the equation solution algorithm (direct or iterative)

At the heart of structural analysis (consuming the lion's share of solution time) is a method to solve large systems of matrix equations which approximate the governing differential equations whether static or dynamic (eigensolution). These equations involve matrices which are predominantly real, positive definite symmetric and sparse. After initial experience with simple iterative methods, the Structures community has migrated through a series of increasingly more effective (faster) solution methods:

Linpack (dense) → Skyline → Banded → Variable-band → Sparse (order all) → Sparse (order subset)

Langley's equation solution research has led to a world-class General-Purpose Solver (GPS), an extension to the Sparse (order subset) algorithm which solves large systems of matrices with any/all of the matrix characteristics listed above. About 10 years ago (1989), a large application was a Shuttle-SRB analysis (54,870 equations) took 14 minutes (13 seconds with Langley's fastest algorithm) on the fastest Cray computer ($4M). Today the same analysis takes 3.4 seconds on an SGI Origin2000 system ($0.3M). Even more important, a moderate size aircraft analysis (552,000 equations) took 53 seconds on the Origin2000. Thus, we can project that ten years hence, complex aircraft and spacecraft structural analyses involving 5-10 million equations will take less than 1 minute.

Electromagnetic and Acoustic applications are similar to structures except that their matrices are generally of COMPLEX data type and sometimes non-symmetric. Traditionally, algorithms to solve Electromagnetic and Acoustic applications have been iterative (time consuming), but lately the trend seems to be to adopt faster direct methods such as NASA's GPS solver (http://transit.larc.nasa.gov/csb-www/GPS.html) as larger memories are available.

Unstructured-grid solvers for viscous flows are now at a state of development nearly on par with structured-grid solvers in terms of accuracy and convergence rate. The large CPU and memory requirements are their drawbacks. However, the advent of large parallel processors with large aggregate memory has enabled calculations to be carried out that were heretofore impossible.

Current algorithmic research is largely focused on improving the convergence rate. Multigrid methods are widely used to accelerate convergence of the time-dependent equations to a steady state. "Textbook multigrid" should allow convergence on the order of a few multigrid cycles. However, when applied to the Navier-Stokes equations and realistic configurations, 500 or more multigrid cycles are required, even by the best solvers. Work is ongoing to understand, and correct, the disparity

between the theoretical rates and those observed in practice. If successful, "textbook multigrid" in a production CFD code could result in a 1-2 order reduction in compute time, which may be leveraged with other means of compute-time reduction, such as parallel processing.

Future algorithm research in all the above disciplines on the cutting-edge applications will involve algorithms to harness the power of multiple processor computers. It is likely that just as the structures community has refined methods resulting in very fast direct methods, the same will occur in Electromagnetics and Acoustics and then by Fluid Mechanics whether techniques such as the Lattice-Gas Automata (LBA) and Lattice Boltzman Equation (LBE) [Luo 1998] methods are effective or not. The advent of very large memories and multiple processors will continue to drive down the solution time in the future even faster than they have in the past 10 years.

For traditional, i.e. PDE based methods, additional work will be required to develop latency-tolerant methods. Optimization methods will need a even stronger focus, particularly to make use of large numbers of processors. One such focus is likely to be on the simultaneous analysis and design methods (SAND) that blend the state and design variables in one vector that defines a single space. The state solution and optimal design are searched for simultaneously in that space.

Among the non-traditional methods, such as the LGA and LBE methods, much more work is required to extend the range of applicability of the methods, and to demonstrate solution accuracy.

4.4 Non-PDE based analysis and optimization

Most of the engineering computing experience to date, and certainly almost all of the LaRC experience, has been underpinned by mathematical modeling based on the partial differential equations. That was usually adequate for most of the vehicle physics where continuous functions and variables commonly occur, but will not be sufficient when that physics is combined with the life-cycle considerations. In that combination one will have to analyze phenomena that are partially continuous and partially discrete, and optimize in a space of continuous variables as well as discrete choices. Some of these phenomena have to be simulated as chains of discrete events in time, e.g., the transactions for material supplies and product deliveries that occur between the prime and sub-contractors.

As an example of a mixed, continuous-discrete problem, consider an aircraft wing that may be made of a conventional riveted sheet-metal or of bonded composites. Either case poses an analysis problem whose state space is divided by the discrete onset of skin buckling into two subdomains, each of the two being internally continuous. In the design space, however, the two types of constructions offer a discrete choice between two dissimilar manufacturing methods. The natural

approach, of course is to eliminate the continuous variables by the conventional analysis and optimization methods and handle the remaining discrete parts by specialized methods and human judgment.

Because design with the life-cycle considerations is replete with choices such as the above, the problem is combinatorial explosion. Fortunately, the large number of concurrently operating processors in synergy with intrinsically discrete methods such as Genetic Programming may keep that explosion under control.

4.5 Human interface with emphasis on visualization

It is said people receive about 90% of information through the sense of vision. Therefore, the focus herein is on visualization, although it is acknowledged that the use of the sense of hearing in the human interface is on the rise, and the sense of touch begins to be utilized as well. Massively parallel processing is seen both as the source of enormous volumes of data to be visualized and as a means by which to achieve an effective visualization. This was one of the reasons why the Department of Energy and National Science Foundation have recently held a series of joint workshops [Smith and Van Rosendale 1998] on manipulation and visualization of large scientific data sets.

The workshop deliberations reflected the dramatic increase in the volume and complexity of scientific data being produced and stored for analysis. However, current tools for graphics and visualization have lagged behind that increase. The workshop series final report examines the state of the art and provides the technology roadmaps of the various components required for visualization of scientific data [Smith and Van Rosendale 1998]. They also characterize the requirements of the scientific community and provide a detailed list of challenges both in hardware and software technologies that need to be overcome to meet the needs of the scientific community. To quote one of the findings of the report [Smith and Van Rosendale 1998]:

> "The development of scientific data manipulation and visualization capabilities requires an integrated systems approach and solution. Such a system must include the end-to-end flow of data from generation to storage to interactive visualization, and must support data retrieval, data mining and sophisticated interactive presentation and navigation capabilities."

Even though the DVC report focuses on applications of interest to DOE and NSF, there is enough overlap to make the findings relevant to NASA needs.

4.6 Impact of distributed computing

The design of aerospace systems, as evidenced by the user scenarios discussed in Section 2 and detailed in the appendix, are becoming increasingly complex requiring teams of discipline experts to cooperate with each other in the solution process. Recent advances in distributed computing environments will allow scientists and engineers to bring disparate and multiple resources including data bases, hardware instruments, computing units and visualization engines, together to solve the larger problem at hand. Such environments will allow geographically distributed teams of users to collaborate on a single task thus making them more effective and productive. Such collaboration will be possible throughout the design process including the initial design, the simulations and the analysis of the results. Thus, for example, engineers from different disciplines could collaboratively steer the simulation to produce better results faster. Also, teams of engineers could use a collaborative tool in order to simultaneously access and visualize the results of the simulations. Similarly research being done in the arena of intelligent agents will allow users to utilize the capabilities of the underlying systems in a more intelligent and effective manner.

Distributed collaborative environments are fundamentally different from the current environments being utilized by engineers and will require not only the development of the underlying software technologies to support the infrastructure but also a change in the manner in which the engineers use these tools. However, it is clear that such environments can engender dramatic improvements in the overall design cycle by allowing the engineers to more effectively use the underlying resources.

5.0 Findings and Recommendations

This section summarizes the UCT findings and recommendations based on the information and discussions reported in the preceding sections.

5.1 Findings

- Acting within its time and resource constraints, the Ultrafast Computing Team has assessed the range of pertinent information regarding engineering design computing
- evaluated the technical requirements required for engineering computing using User Scenarios focused on an advanced aircraft and reusable launch vehicle applications
- reviewed the trends in computer hardware and software
- identified actions to be taken in order to exploit in engineering computing the opportunities likely to be created by the above trends
- recommended formation of a team at LaRC to implement the above actions.

Engineering computing needs vs. Computer Technology progress

The times to solution in many engineering problems that arise in context of design far exceed of what is required to meet the current and future advanced vehicle design needs. They also fall far short of the goals derived from the NASA 3 Pillars strategy.

Rapid progress of the general purpose computer technology will continue toward a heterogeneous environment in which the spectrum of computers will range from single processor machines to multiprocessor ones. The number of processors in the latter will likely to increase into hundreds of thousand, and may even exceed a million. High speed, intelligent networks will unify the heterogeneous computer architectures, including special purpose computers dedicated to a specific task, to present a single virtual computer to the user.

To exploit the multitude of new computer architectures, and to exert influence on development of these architectures, the present computing tools in engineering analysis and synthesis will have to be tailored to fit the architecture(s) most suitable to the class of problems these tools are for. Wherever a fit cannot be achieved but potential gains are substantial, new paradigms of computing will have to be invented. The roots of the new paradigms may have to go deep into the underlying physics and foundations of those problems where the needs are acute for radically faster solutions. It is apparent that doing that well will require long lead times and large investment.

How to get engineering computing to ride the wave of the future in computer technology

The engineering computing market is small relative to that in business and entertainment. Therefore, it constitutes a niche where the Government seed money might make a real difference.

The situation calls for the following actions to be taken at LaRC:

1) In the interdisciplinary arena, one should continue to

   - monitor, understand the new computer hardware and software technologies and architectures
   - develop an understanding of the capabilities that are likely to be delivered by the commercial development regardless of the Government actions
   - influence development of the new computer hardware and software technologies and architectures
   - develop understanding of the match between various types of engineering computing jobs and various computer architectures, and the match frequencies
   - formulate the need for new developments at the integrating framework level and at the disciplinary level in particular discipline
   - formulate standards and requirements as needed by the tool integration, MDO environment, and the new architectures
   - develop methods for effective utilization of the system analysis and MDO for various classes of the new architectures, taking into consideration the computing load balancing among the processors
   - recommend long term investment strategy based on the above information
   - foster & coordinate disciplinary developments and application projects
   - facilitate education and training

2) In each disciplinary domain, one will need to

   - commit to gearing-up to the exploitation of new computer architectures in hardware and software.
   - reexamine and restructure the disciplinary algorithms, and to develop new paradigms where needed, accounting fully for MDO

- formulate local disciplinary standards and requirements compatible with the ones established in the interdisciplinary arena
- develop and validate the restructured algorithms and the new paradigms, implementing the standards and requirements

3) In both the interdisciplinary and disciplinary arenas, one should
   - Validate new algorithms
   - Demonstrate benefits
   - Foster technology transfers to industry users

5.2 Recommendations

The above actions may be carried out by an interorganizational, project-based, team formed to foster the incorporation of advanced high performance computing into aerospace design processes.

The character and charter of the team, called Computational Engineering Team (CET), would be:

1. An expanded version of the CAS team. The CAS team model is proven and well suited to this kind of activity. The funding should be drawn from the programs such as HPCCP/CAS, ASPO, and ISE, as well as the base research.

2. CET would have a core to perform the interdisciplinary functions described in #1 and #3 above. It would fund and coordinate the disciplinary organizations to perform the disciplinary functions described in #2 and #3 in the preceding section.

3. In performing the above functions, CET would:

3.1 Monitor and investigate current and projected advances in CPU speed, networking, storage, system software, and applications support software (e.g. visualization)

3.2. Support the application of current high performance computing techniques to legacy discipline codes. Discipline-oriented branches would receive either funds or contractor support for the conversion of their codes. More concentration needs to be applied to everyday applications, not just grand challenges currently named in the CAS Program.

3.3. Organize and fund demonstration projects, like the current HSCT4.0, that cross discipline boundaries.

3.4. Support basic research into new, innovative methods that exploit the advances in IT (a la the current NRA grants). The research could be performed at LaRC, other government facilities, industry, and academia. Demonstrate the new methods as replacing or augmenting legacy codes.

3.5 Implement major elements of the ISE Program

3.6. Advocate, to the IT community, the development of capabilities that would be of particular benefit to engineering applications

6.0 References

Anderson, T., Dahlin, M., Neefe, J., Roselli, D., Wang, R., Patterson, D., "Serverless Network File Systems," Technical Report UCB-CSD-98-986, 1998.

Anon. The National Technology Roadmap for Semiconductors; 1997 edition (published in January, 1998); Semiconductor Industry Assoc. (S.I.A.)

Coyne, R. A., Hulen, H., Watson, R. W., "The High Performance Storage System", Proceedings of Supercomputing 93, Portland, November 1993.

Feiereisen, William: Information Power Grid (IPG), Presentation to the IPG Technical Meeting, June 1998, NASA Ames Research Center

Gibson, G., Nagle, D., Amiri, K., "A Case for Network Attached Secure Disks," Technical Report CMU-CS-96-142, June 1996.

Grochowski, E., Hoyt, R., "Future Trends in Hard Disk Drives," *IEEE Transactions on Magnetics*, 32(3), May 1996.

Luo, Li-Shi: The Future of Lattice-Gas and Lattice Boltzmann Methods. Proceedings of the ICASE/LaRC/NSF/ARO Workshop on Computational Aerosciences in the 21st Century, April 22-24, 1998, Hampton, Virginia. (In preparation)

Mavriplis, D. J.: "Three-Dimensional High-Lift Analysis Using a Parallel Unstructured Multigrid Solver", AIAA-98-2619, June, 1998.

Moore, R., Prince, T.A., and Ellisman, M., "Data-Intensive Computing and Digital Libraries," *Communications of the ACM*, 41(11), November 1998, pp. 56-62.

Patterson, D. A., Gibson, G. A., Katz, R. H., "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Technical Report UCB-CSD-87-391, 1987.

Smith, Paul H. and Van Rosendale, John, eds.: "Data Visualization Corridors: Report on the 1998 DVC Workshop Series" , California Institute of Technology, 1998.

Sobieszczanski-Sobieski, J.; and Haftka, R.T.: Multidisciplinary aerospace design optimization: survey of recent developments; *Structural Optimization*, a Springer Verlag journal, Vol. 14; No. 1; August 1997.

Stevens, R., Woodward, P., DeFanti, T., Catlett, C., "From the I-WAY to the National Technology Grid," Communications of the ACM, 40(11), November 1997, pp. 51-60.

Talagala, N., Asami, S., Anderson, T., Patterson, D., "Tertiary Disk: Large Scale Distributed Storage," Technical Report UCB-CSD-98-989, 1998.

Tristram, C.: The Big Bad Bit Stuffers of IBM, MIT Technology Review, July-August 1998, pp.45-51.

von Eicken, T., Vogels, W., "Evolution of the Virtual Interface Architecture," IEEE Computer, November, 1998 pp. 61-68.

Appendix

A.1 User Scenario 1: A Supersonic Transport Case

An emerging practice aimed at ensuring software systems do what the customer intends them to do is becoming known as "Use Case Scenarios". A Use Case is a textual description of the process the customer is asking the software system to do. The Use Case can include graphical representations of processes that enhance the text. The output of the Use Case provides a common platform for both the software developer and the customer to interact on. Once the objective is understood, the analysis of the requirements inherent in the Use Case can commence. From here, a software development process coupled with appropriate prototypes (Spiral Development) can take over.

The Use Case is a process for the customer and the developers to communicate. It is through this mechanism that an architecture will be developed, validated and extended. The following example is a Use Case for the IPG.

USE CASE NAME: Whole Aircraft
USE CASE NUMBER: 101
AUTHOR(s): NASA LaRC CAS Team and LeRC IPG Team
CREATION DATE: 970904
REVISION DATE:

ABSTRACT:

This use case describes how a User would execute an optimization of an elastic airframe with engines attached using the IPG across the US with Multi-Company participation and viewing. The case involves codes in aerodynamics, structures, propulsion, aircraft performance, and optimization, all integrated and executing in both analysis and optimization modes.

ASSUMPTIONS:

0) A supersonic transport aircraft is at an early Preliminary Design stage. The project is conducted by a partnership of several major, geographically dispersed companies. The week of September 12, 2011 is a "downselect" period when the partnership must choose between two proposed alternative configurations A and B.

Configuration A is a tailless canard with two engines X. Configuration B is a conventional configuration with four engines Y. Before the week begins, each configuration had been defined including airframe detailed design variables (e.g., cross-sectional structural dimensions, airfoil camber) and major design variables (e.g., wing sweep angle, aspect ratio) but had not been optimized yet. As well as propulsion detailed design variables (e.g. Compressor Pressure Ratio, # of

compressor stages and blade rows , and major design variables (e.g. thrust, weight, specific fuel consumption, engine and spool RPM, fuel flow rate) but had not been optimized yet either. The performance objective to be maximized is the range for a given payload under all the disciplinary constraints.

The partnership has a week: to bring A and B to the same level of optimality, to evaluate the results including their economics, to select A or B.
The choice will hinge on small differences of large numbers in multidisciplinary trades, e.g., drag vs. weight, hence a high fidelity of analyses is important.

1) The distributed (geographically dispersed) simulation of the aircraft aerodynamics, structures, propulsion, life-cycle economics, and aircraft performance is comprised of a known number of codes. The codes are equipped in a sensitivity analysis, that is they can compute derivatives of output with respect to input.

2) In addition to disciplinary codes there are
   - a library of optimizers (codes that search design spaces).
   - utilities to assist in optimization, e.g., response surface generators, neural net codes.
   - an agreed-upon MultiDisciplinary Optimization (MDO) method implemented as a script to sequence the codes.

3) There is a separate engineering group for each of the following: aerodynamics, structures, propulsion, life-cycle economics, and aircraft performance. Each group belongs to a different company. The aircraft performance group acts as a "chief designer" to manipulate major design variables of both the airframe and engine as needed toward meeting the design objectives.

4) Each engineering group has an autonomy over their codes for analysis and local optimization, and over formulation of their constraints.

5) One additional company is viewing the analysis but not directly participating.

6) All output and input is seen by all parties at the same time.

7) Security is in place to prevent unexpected intrusion into the simulation while allowing data to flow freely.

8) The participants use a mix of workstations and supercomputers, adding up to tens of thousands of simultaneously accessible processors with memories to match.

9) Companies are spread across the US

ACTORS:

1) Engineers at the partnership companies
2) Viewer/customer at another site

PROCESS OVERVIEW:

0) The site of the aircraft performance group is the prime coordinator of the design optimization.

1) The processes for A and B are being conducted independently with a degree of concurrence at the discretion of the partners.

2) Each process is an iteration alternating between the analyses and optimizations conducted concurrently within each group specialty domain using local design variables, and the system-level optimization executed by the performance group when returns from the groups are available.

3) The process preserves couplings among the disciplines by routing output to input between the codes so that any local or major design change may propagate throughout the system. Examples of the couplings are: aerodynamics-structures in aeroelasticity, propulsion-aerodynamics-structures, all disciplines-life-cycle economics.

4) The type of information produced by the groups and the detailed logic of the process are prescribed by the agreed-upon MDO method.

5) The process is interactive both at the group level and the system level so that the windows into the process progress are open and decision points are provided for. The actors work in a "mission control room" environment with numerous displays and means to communicate with each other. Using these means they:

- ask "what if" questions, each of which might initiate additional analyses and optimizations.
- accept or reject the process output,
- intervene into the process by adding and deleting the design variables and constraints, modifying the assumptions, and generally imposing their judgment on the process as they see fit. Imposition of judgment assures that the subjective, non-quantifiable, and discrete-choice aspects of design are fully facilitated.

6) When A and B are both optimized to reach their potential, the partners compare both, drawing to any depth of detail on the data produced and archived in the process, and asking additional questions that may trigger repetitions of some portions of the process. They make their choice between A and B when they feel they have all the information to support it.

COMPUTING REQUIREMENTS:

1) CFD, NS-level, entire configuration analysis with derivatives of output to input turn-around - under 1 minute

2) Structural analysis, FEM-level, entire airframe of about 100K EDOF, 1000 loading cases, with derivatives of output to input turn-around - under 10 sec.

3) Propulsion, CFD, NS-Level 3Dimensional Transient simulation of Full engine with Numerical Zooming and MD coupling within 1 minute to match surrounding simulation optimization time.

4) Simulation of dynamic phenomena, e.g., the vehicle handling, closely to the real time scale.

5) Other disciplines turn-around - nearly instantaneous.

6) Visualization and Virtual Reality capability to support real time movies to display dynamic phenomena: at least 24 frames/sec on high resolution screens.
Implication: tens of thousands processors available and engineering analysis codes capable of exploiting that parallelism.

7) Flexibility to "shift gears" between codes of different fidelity to trade computing cost for accuracy, and to change from one MDO method to another.

FREQUENCY/DURATION OF INVOCATION:

1) Complete the round of concurrent disciplinary operation in under 60 minutes.
2) Complete the collaborative system-level optimization and assessment of the results before committing to the next round of disciplinary optimizations in under 90 minutes.
3) Sustain a pace of 4 to 5 cycles in a working day, each cycle entailing #1 & 2 above.
4) Obtain answer to a minor "what if" question in less than 10 seconds
5) Obtain an answer to a major "what if" question in less than 2 minutes.

PARTICIPATING OBJECTS:

1) Data objects for analysis and optimization.
2) Specific disciplinary codes.
3) Optimization codes
4) Script implementing an MDO method.
5) Utility codes to visualize and to control the process

A.2 User Scenario 2: A Reusable Launch Vehicle Case

An emerging practice aimed at ensuring software systems do what the customer intends them to do is becoming known as "Use Case Scenarios". A Use Case is a textual description of the process the customer is asking the software system to do. The Use Case can include graphical representations of processes that enhance the text. The output of the Use Case provides a common platform for both the software developer and the customer to interact on. Once the objective is understood, the analysis of the requirements inherent in the Use Case can commence. From here, a software development process coupled with appropriate prototypes (Spiral Development) can take over.

The Use Case is a process for the customer and the developers to communicate. It is through this mechanism that an architecture will be developed, validated and extended. The following example is a Use Cases for the Fast, Efficient Design Tools.

USE CASE NAME: Reusable Launch Vehicle
USE CASE NUMBER: 203
AUTHOR(s): NASA LaRC FED Team
CREATION DATE: 981105
REVISION DATE:

ABSTRACT:

This use case describes how a User would carry out a particular operation in the design process of an RLV with multi-company participation and viewing. The design process includes a wide range of the mission architectures and the entire life-cycle of the vehicle. The range of missions includes unmanned payloads as well as manned ones. Therefore, although the vehicle control is automatic but it provides for a human pilot override option. The case involves codes in life-cycle, component and payload packaging, aerodynamics, including rarified atmosphere reactive flow, structures, including hot structures, propulsion, vehicle performance, including trajectory, heat transfer, and optimization, all integrated and executing in both analysis and optimization modes. The case was selected to require switching between the degree of detail characteristic for the Detailed Phase and the Conceptual Phase of the design process. Such switching is often necessary when new information is generated at the detail level that might suggest revisions of the major configuration decisions that had already been made early in the design process.

ASSUMPTIONS:

0) An RLV is at an early Preliminary Design stage. The project is conducted by a partnership of several major, geographically dispersed companies. The week of September 13, 2020 is a "downselect" period when the partnership must choose between two proposed alternative configurations A and B.

Configuration A is a Single-Stage-To-Orbit, Vertical Take-Off, Horizontal-Landing, Hybrid-Airbreathing vehicle. Configuration B is a Horizontal Take-Off with Electromagnetic Assist, Expendable Booster, Horizontal Landing vehicle. Before the week begins, each configuration had been defined including detailed design variables (e.g., cross-sectional structural dimensions, wing airfoil, details of propulsion) and major design variables (e.g., wing sweep angle, aspect ratio, body major dimensions and proportions, thrust-to-weight ratio, manufacturing and operations variables) but had not been optimized yet.. The performance objective to be maximized the return on investment over the life of the vehicle, including the assumptions of 10 years and 36 launches per year. The return on investment must account for the life-cycle economics.

The partnership has a week: to bring A and B to the same level of optimality, to evaluate the results including their economics, to select A or B.

The mission profitability, and the corresponding ROI, will hinge on small differences of large numbers in multidisciplinary trades, e.g., structural weight vs. structure cost. This trade-off must account for the structural weight effect on the propulsion required and its influence on the vehicle gross weight and total cost, hence a high fidelity of analyses is important.

The task is made more complicated by the discovery in the aeroelastic analysis that in Configuration A the structural displacements of the vehicle nose is excessive in a way it displaces the shock wave impinging on the engine inlet. An attempt to fix the problem by structural optimization led to excessive structural weight penalty, therefore, an alternative of relocating the engine to make it less sensitive to the nose-generated shock wave is under consideration. However, to assess fully the benefit of that alternative one has to return to the Conceptual Stage in order to modify the configuration.

1) The distributed (geographically dispersed) modeling of the life-cycle, component and payload packaging, aerodynamics, including rarified atmosphere reactive flow, structures, including hot structures, propulsion, vehicle performance, including trajectory, heat transfer, and optimization comprises a known set of codes. The codes are equipped with sensitivity analysis, that is they can compute derivatives of output with respect to input.

2) In addition to disciplinary codes there are available
   - a library of optimizers (codes that search design spaces).
   - utilities to assist in optimization, e.g., response surface generators, neural net codes, data base(s).
   - an agreed-upon MultiDisciplinary Optimization (MDO) method implemented as a script to sequence the codes.
   - software framework (commercially available with maintenance) that ties the codes and utilities in a unified tool users can interact with. The tool can be commanded to execute the selected MDO script.

3) The modeling and design-decision making is distributed over the participating companies. The mission architecture group acts as the "chief designer" to manipulate major design variables of both the airframe and propulsion and those of the life-cycle as needed toward meeting the design objectives. This group identifies the major "what if" questions and design variables. Where appropriate it delegates to the specialists determination of the answers to these questions and the effects of the design variables.

4) Each engineering group has an autonomy over their codes for analysis and local optimization, over formulation of their constraints, and over their local optimization methods .

5) One additional party is monitoring the design process by observing the data generation and the data evolution without directly intervening in the process.

6) All output and input can be seen by all parties may look at all the inputs and outputs at the same time.

7) Security is in place to prevent unexpected intrusion into the process while allowing data to flow freely.

8) The participants use virtual computers, adding up to tens of thousands of simultaneously accessible processors with memories to match.

9) Companies are spread over the world.

ACTORS:

1) Engineers at the partnership companies
2) Viewers/customers/regulatory agency monitors at other sites.

PROCESS OVERVIEW:

0) The site of the mission architecture group is the prime coordinator of the design optimization.

1) The processes for A and B are being conducted independently with a degree of concurrence at the discretion of the partners.

2) The key feature of the process is the way in which it enables the engineers to exercise to the fullest their ingenuity to deal with the numerous, discrete choices and judgment-dependent, difficult-to-quantify, issues that arise in design. To support such high-level decision making, the system automatically analyses and optimizes all the detailed design variables where the relations are continuous and the optimizations may be executed without human intervention. The system has enough intelligence built-in to keep such optimizations within the bound of assumptions and to alert engineers when the results "bump" against those bumps. The system has also operational flexibility to enable the mission architecture group to delegate detailed investigation to generate information needed at the mission architecture level. The detailed investigations (analyses & optimizations) are performed at the appropriate detail level, off-line as far as the mission architecture is concerned.

3) Each process is an iteration alternating between the analyses and optimizations conducted concurrently within each group specialty domain using local design variables, and the system-level optimization executed by the performance group when returns from the groups are available.

4) The process preserves couplings among the disciplines by routing output to input between the codes so that any local or major design change may propagate throughout the system. Examples of the couplings are: aerodynamics-structures in aeroelasticity, propulsion-aerodynamics-structures, all disciplines-life-cycle economics.

5) The type of information produced by the groups and the detailed logic of the process are prescribed by the agreed-upon set of standards and protocols that reflects an MDO method selected. Engineers are enabled to intervene into the process logic.

6) The process is interactive both at the group level and the system level so that the windows into the process progress are open and decision points are provided for. The actors have access to a "mission control room" environment with numerous displays and means to communicate with each other. Using these means they:

- ask "what if" questions, each of which might initiate additional analyses and optimizations.
- accept or reject the process output,

- intervene into the process by adding and deleting the design variables and constraints, modifying the assumptions, and generally imposing their judgment on the process as they see fit. Imposition of judgment assures that the subjective, non-quantifiable, and discrete-choice aspects of design are fully facilitated.

7) The software supporting the process allows a high degree of "computing agility", a capability of switching easily between the models of various levels of fidelity and resolution. This enables one to move back and forth over the entire spectrum of the design phases, Conceptual to Detailed and back again, as the need arises.

8) The Product Data Model includes a master geometry model and a multitude of models that simulate various aspects of design and the life-cycle, all dependent on the master model. Whenever the master model changes, all the dependent models follow automatically. These models are smart enough to recognize when the master model change may be pushing them beyond the constraints and to automatically adjust or to alert the engineer.

9) When A and B are both optimized to reach their potential, the partners compare both, drawing to any depth of detail on the data produced and archived in the process, and asking additional questions that may trigger repetitions of some portions of the process. They make their choice between A and B when they feel they have all the information to support it.

FUNCTIONALITIES REQUIRED:

0) **Assumption**: tens of thousands processors available and engineering analysis codes capable of exploiting that parallelism.

1) CFD, NS-level, including rarified atmosphere reactive flow, entire configuration analysis with derivatives of output to input turn-around, under 1 minute

2) Structural analysis, FEM-level, entire airframe of about 100K EDOF, 1000 loading cases, with derivatives of output to input turn-around, including heat transfer and hot-structures, under 10 sec.

3) Propulsion, including solid booster, electromagnetic propulsion, and airbreathing propulsion, 1 minute to match surrounding simulation optimization time.

4) Simulation of dynamic phenomena, e.g., the vehicle handling, close to the real time scale.

5) Modeling of the life-cycle elements other than the vehicle physics, taking into account the predominantly discrete and statistical nature of these elements in

elapsed times comparable to those achieved for the vehicle physics modeling. This includes the analysis of cost and revenue necessary for assessment of the ROI.

6) Other disciplines turn-around - nearly instantaneous.

7) Modeling of the management of the design process in elapsed times comparable to those achieved for the vehicle physics modeling.

8) Use of intelligent agents for continual monitoring of the process, information gathering, warnings about unexpected conditions arising in the process, and making routine decisions.

9) Visualization and Virtual Reality capability to support real time movies to display dynamic phenomena: at least 24 frames/sec on high resolution screens.

10) Immersive capability enabling virtual walk throughout the mission architecture and its execution in time; peeling off the external parts of the vehicle to look inside, and invoking analysis by touching the part of the vehicle and pointing to the computing tool to be applied to that part.

11) Flexibility to "shift gears" (computing agility) between codes of different fidelity to trade computing cost for accuracy, to return from the Detailed stage to the Conceptual one as the need requires, and to change from one MDO method to another.

FREQUENCY/DURATION OF INVOCATION:

1) Complete the round of concurrent disciplinary operation in under 60 minutes.
2) Complete the collaborative system-level optimization and assessment of the results before committing to the next round of disciplinary optimizations in under 90 minutes.
3) Sustain a pace of 4 to 5 cycles in a working day, each cycle entailing #1 & 2 above.
4) Obtain answer to a minor "what if" question in less than 10 seconds
5) Obtain an answer to a major "what if" question in less than 2 minutes.

PARTICIPATING OBJECTS:

1) Data objects for analysis and optimization:
    - Mission Architecture
    - RLV
    - RLV Life-Cycle.
2) Specific disciplinary codes, including the life-cycle and its economics..
3) Optimization codes
4) Script implementing an MDO method.
5) Utility codes to visualize and to control the process

6) Software framework to tie codes and utilities into a single tool, spanning the geographically dispersed sites, and accommodating a variety of physical computing platforms.

A.3: Real Time Simulation and Animation.

Analysis, as well as visualization, divides into two modes or classes of study and presentation. These are variously described as Static and Dynamic - or alternatively, Steady State and Time Dependent. Static (steady state) can be characterized as a single solution per analysis initial conditions. Dynamic (time dependent) generally requires many solutions because usually it is necessary to discretize the time dimension (as well as the physical dimensions). Various methods may be employed to display the static solution. In contrast, the dynamic solution is most often displayed as an animation, consisting of a number of time steps, displayed as a movie. The dynamic case places a much greater calculation load on the computer. There is no simple answer to the ratio of computations for dynamic versus static.

As an initial example, for 24 images per second and two minutes of animation, 2,880 solutions (24*120) will be required. For this simple example, the demands on the computer would be nearly 3,000 times as many operations. It is not generally that simple. Each frame of the dynamic simulation is mathematically equivalent to the single steady state solution. For the static solution, let assume that it takes 500 iterations to converge. For a dynamic solution, each succeeding result is close to the previous. As a result the steps to converge may not require any more operations than one iteration. Thus the 2,880 frames could take only about six times the computing effort as the static case. The time steps for good animation may not correspond to the time step size needed for good convergence or an appropriate animation rate. Frequently it becomes necessary to calculate many more solutions than are displayed. This might mean that four time steps are used for each frame, with three thrown away for each one saved. Then in this example, the dynamic solution would require about 2,400 times as much compute time as the static.

The scenarios place demands upon the computer for the static problems that are severe or beyond present computers to produce solutions within minutes. Thus the dynamic cases are well above this and become critical in sizing the computer system. (The storage of the solutions for a series of dynamic animations becomes exceeding large also.) If this were the whole story, doing dynamic simulations would become out of the question.

Currently, however, the calculations can be done in a batch mode in a three step process and then the results shown in "real time". This method would have to be reviewed for the scenarios because of the demand for interaction, but it is likely that instead of a speed up ratio of for example, of from two hours to 24 per second and one frame to 3,000 frames calculated (2*60*60*24*3,000=518,400,000) a solution per minute would reduce the computer rate by a factor of 1,440 (60*24). There are other factors to consider for the real time aspects of the scenarios.

A CFD static problem, as an example, often involves a pre-processing step of grid generation, a second step of solution calculation, and a third post-processing step to

convert the solution into parameters suitable for visualization. The first step may only be involved once per a given geometry. The second step is done many times for different conditions. The third step is done many times from different aspects for solutions that are deemed "valid". For the dynamic case, an additional step is needed for the animation. This can be done interactively if the viewer can specify the animation viewpoint and model dynamics needed to visualize the simulation. Therefore with careful planning, it may be possible to handle the dynamic situation within the computation limits of a petaflop (10^{15} ops./sec.) system that will easily handle static cases.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1999	Technical Memorandum

4. TITLE AND SUBTITLE
Compute as Fast as the Engineers Can Think!
ULTRAFAST COMPUTING TEAM FINAL REPORT

5. FUNDING NUMBERS
509-10-31-03

6. AUTHOR(S)
R. T. Biedron, P. Mehrotra, M. L. Nelson, F. S. Preston, J. J. Rehder,
J. L. Rogers, D. H. Rudy, J. Sobieski, O. O. Storaasli

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17891

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM-1999-209715

11. SUPPLEMENTARY NOTES
This report is also available in electronic form at URL http://techreports.larc.nasa.gov/ltrs/

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 61 Distribution: Standard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This report documents findings and recommendations by the Ultrafast Computing Team (UCT). In the period 10-12/98, UCT reviewed design case scenarios for a supersonic transport and a reusable launch vehicle to derive computing requirements necessary for support of a design process with efficiency so radically improved that human thought rather than the computer paces the process. Assessment of the present computing capability against the above requirements indicated a need for further improvement in computing speed by several orders of magnitude to reduce time to solution from tens of hours to seconds in major applications. Evaluation of the trends in computer technology revealed a potential to attain the postulated improvement by further increases of single processor performance combined with massively parallel processing in a heterogeneous environment. However, utilization of massively parallel processing to its full capability will require redevelopment of the engineering analysis and optimization methods, including invention of new paradigms. To that end UCT recommends initiation of a new activity at LaRC called Computational Engineering for development of new methods and tools geared to the new computer architectures in disciplines, their coordination, and validation and benefit demonstration through applications.

14. SUBJECT TERMS
computers, analysis, optimization, parallel processing, concurrent processing, aerospace design, computational engineering, high-performance computing

15. NUMBER OF PAGES
54

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL